CSC510 April Report: project 2

Sz Ting Tzeng and Matt Farver and Tam Nguyen

1 Data

1.1 Data Collection

Utilizing Github APIs, we collected our target data from Github for three teams and with following properties:

- Milestones
  - Milestone ID
  - Milestone title
  - Milestone description
  - Created time
  - Due time
  - Closed time
  - Created user
- Issues
  - Issue ID
  - Issue title
  - Created time
  - Closed time
  - Created by
  - Closed by
  - Label
  - Assignee
  - Milestone
- Commits
  - User
  - Timestamps
  - Message
- Comments
  - Comment ID
  - Comment content

- User
- Created time
- Update time

The data was grabbed with a customized gitiable.py and saved into csv files. Apart from mentioned properties, we add more information into dataset for easier operation so all our analysis could be done with local csv files.

1.1.1 Data format and sample- Milestones

id	Milestone id	Milestone title	Milestone Description	Created at	Due at	Closed at	User	Group name
1	1	Jan milestone	Project idea	(time)	(time)	(time)	user1	group1

1.1.2 Data format and sample- Issues

id	Issue id	Issue title	Created at	Closed at	Created by	Closed by	Label	Assignee	Milestone	Group name
1	1	(title)	(time)	(time)	user1	user1	enhancement	user1	(text)	group1

1.1.3 Data format and sample- Commits

sha	User	Time	Message	Group name
(sha)	user1	(time)	(text)	group1

1.1.4 Data format and sample- Comments

id	Issue id	Comment	User	Created at	Updated at	Group name
1	1	(text)	user1	(time)	(time)	group1

1.2 Preprocessing and Anonymization

To keep privacy, user names and group names were anonymized and replaced with string plus sequence number. List the pseudo code as below.

```
function anonymization(user_name):
    if user_name not in User Map:
        Add user_name into Map with value "user" + (length of user_list + 1)
    return Map value with key [user_name]
```

1.3 Data

The summary of our collected dataset lists below:

Milestones	Issues	Commits	Comments
17	118	334	374

2 Milestones

In project 2 with milestones data, we focused on the relationships between milestone-related timestamps, created time and due time and closed time, and ignored text data. To be more specific, we analyzed the state-changed time of milestones and try to reveal possible problems. Basically, a well-structured project specifies goals in each steps and with sufficient description, and the whole project runs with the defined schedule. We believe by analyzing milestone state updates we could take a glance of project management.

2.1 Hypothesis-Question

The question was, did people follow their schedule and close their milestones before deadline? In our hypothesis, if a project is well-managed, the defined milestones should be checked and closed before due date.

2.2 Analysis

We expected to see milestones with stable updates but we found some milestones without close date or due date. Figure 1 displays that only group 2 setup and update their milestones in Github relatively complete. The possible reason for this phenomenon is that people may count on other communication approaches in each group.

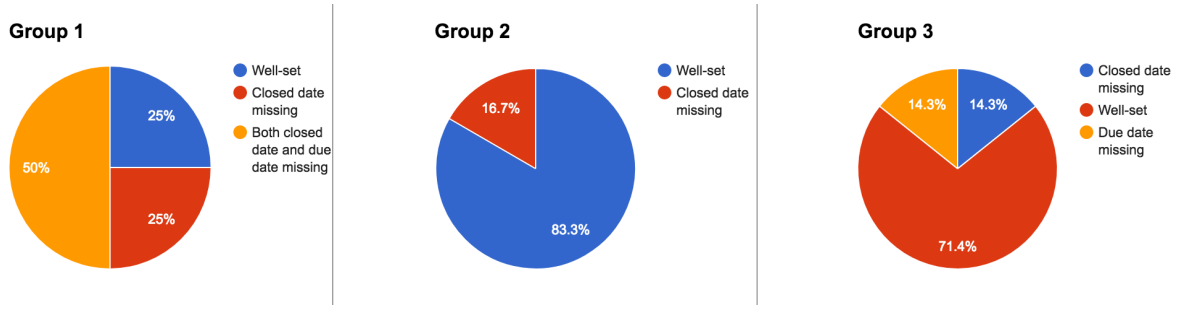

Figure 1. Milestone time management

Figure 2 displays the percentage of current milestone status. Obviously, most milestones are closed after due date or even still open now. Furthermore, with Figure 3, a scatter chart without incomplete data, we found that group 3 closed their milestones long after the due date. Both findings and the anomalies indicate that our assumption that milestones will be closed before due date was failed in project1. A possible explanation is that groups didn't manage their projects with Github. However, in industry poor milestone usage is a huge mistake and may affect project result. Checking milestones in each step could be an efficient warning detector for projects.

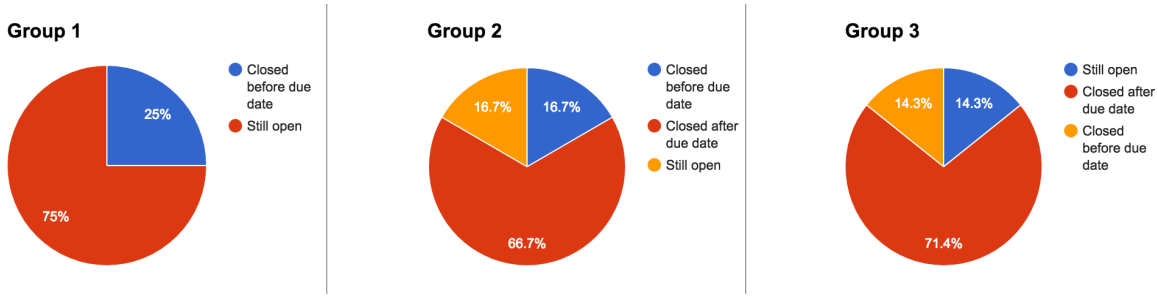

Figure 2. Current milestone status

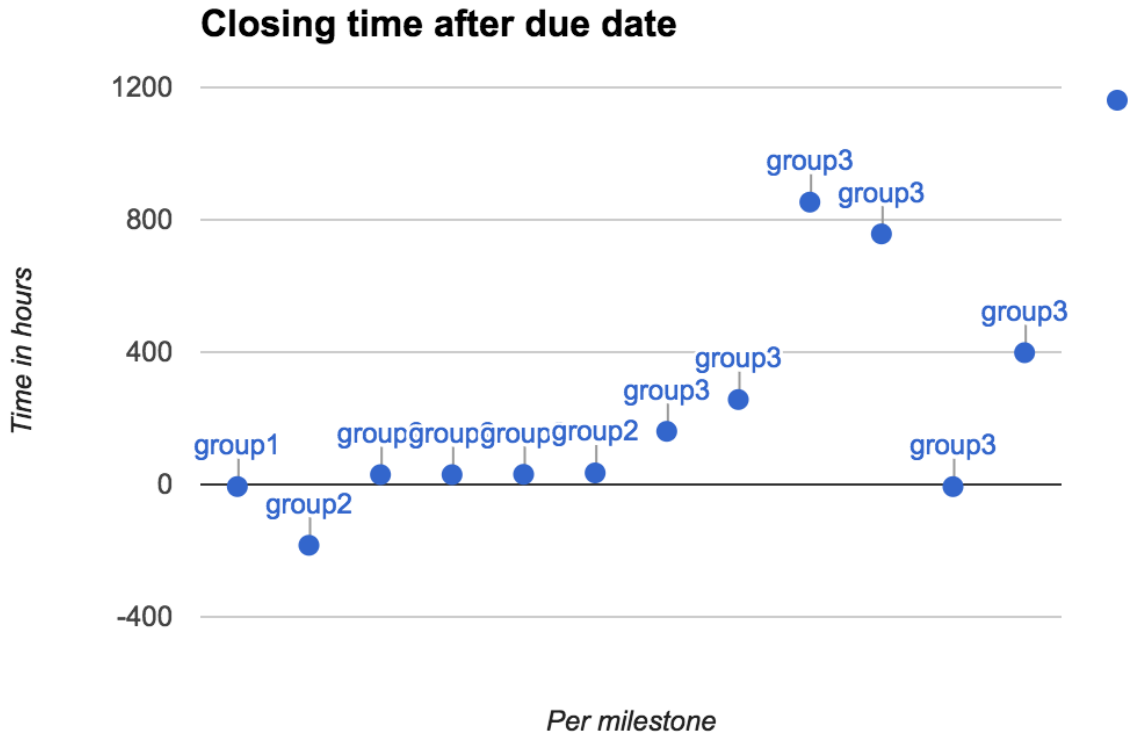

Figure 3. Time after milestones closing

3 Issues

The mining of issues data was focused on time-related and person-related information. Specifically, we analyzed the time lasting of issues, milestone assignment of issues, time-line of issues, issue assignments and the number of issues created or closed by each team members. We expected to identify workload distribution and project progress by analyzing above data.

3.1 Hypothesis-Questions

Our first question on issues was, were issues closed in reasonable time? our hypothesis here was that we had different goal in each month and issues created should not last more than a month.

Our second question was, were issues assigned to specific milestone? We hypothesized that issues should be assigned to milestones in project 1 since we had specific goals in every stage.

Our third question was, were issues closed before due date of the assigned milestone? We hypothesized that issues created as specific units of milestones will be closed before milestone due.

The fourth question was, were issues assigned to each team member relatively equal? Our hypothesis was that the partition should be close to equal among team members.

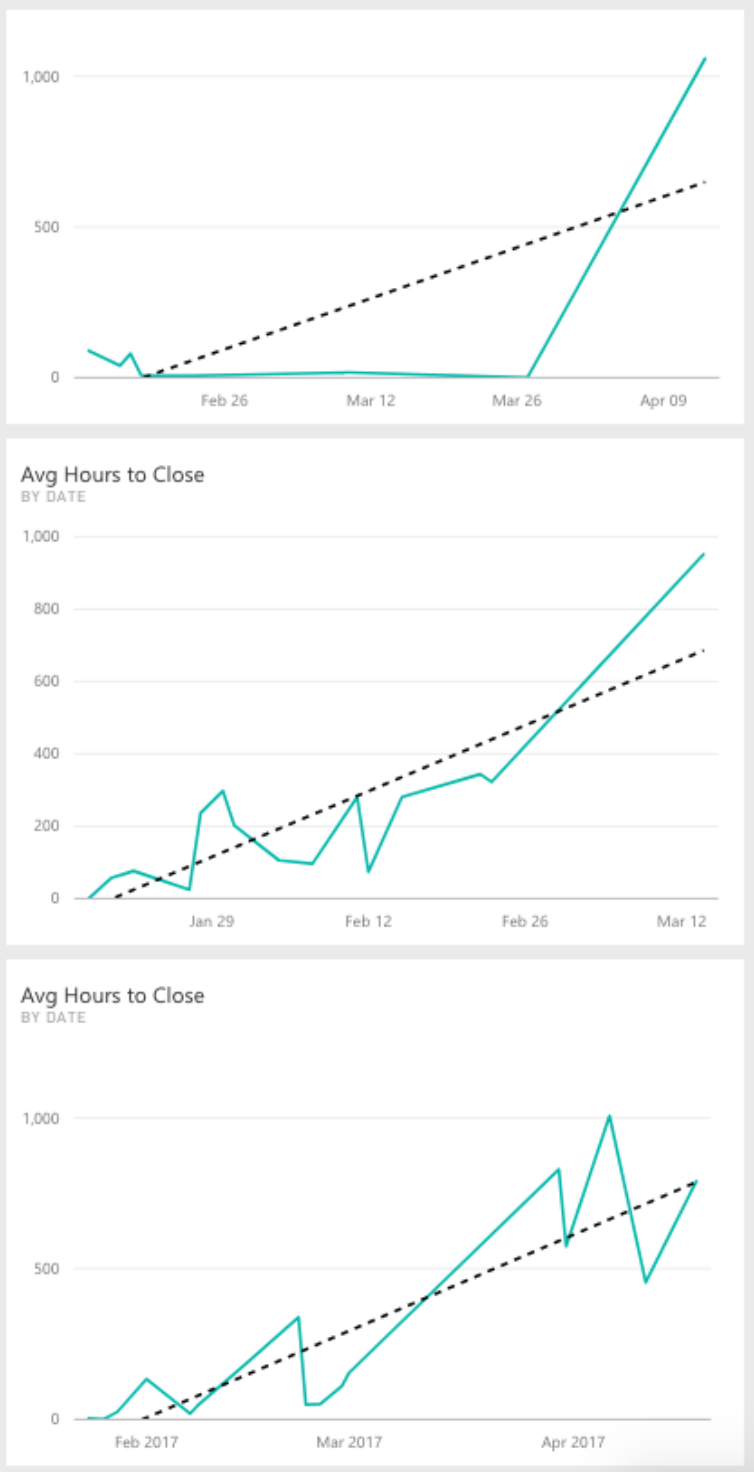

Figure 4. Average hours to close an issue (group H, P, S)

3.2 Anomalies

During the mining, we found some anomalies. First, as Figure 5 displays, group 3 had an issue lasting for 60 days, and after we digging deeper we found that it was an issue still open now.

Second, we found some issues without milestones. After digging in, we ignored some of them in this analysis since they're testing issues created by advisors.

Third, there exist many milestones without due date as Figure 1, and we decided to ignore issues assigned to these milestones

for our third hypothesis.

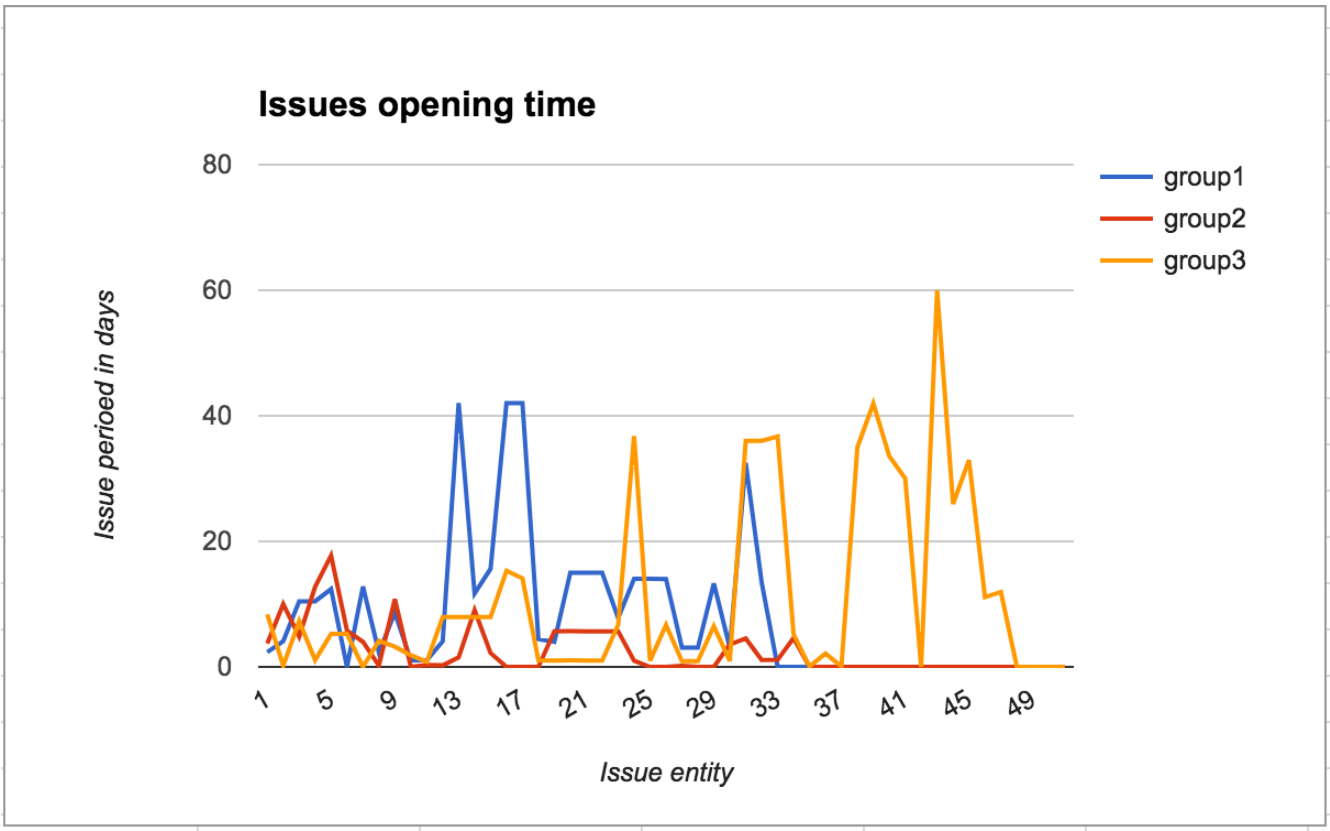

Figure 5. Issue opening time

3.3 Analysis

In Figure 5, both group1 and group3 had issues opened more than 30 days. On the other hand, group2 managed issues as we expected. After diving deeper, group1 had 4 issues and group3 had 10(including the still open issue) opened more than 30 days. In group1, 2 of the long opened issues were not task-oriented issues which may explains why they last longer. However, for those task-oriented issues, the possible explanation is that people didn't rely on Github as a project management tool. Checking issues status regularly could be an efficient waning detector since Not having issues resolved stably may indicates poor communication or lack of management.

Figure 7 shows that some issues were not assigned to milestones, and after we looked into data, we found group1 and group3 had one or two issues created for internal meeting schedule. On the other hand, both group2 and group3 had issues related to stage goals but without milestone assigned. Not being assigned milestones may be reasonable for optional issues, but for critical issues that might be a warning.

According to Figure 6, 86% issues of group1 closed before due dates, and 60% for group2 and 46% for group3. This finding to some extent validates our second hypothesis and for those exceptions, we think how people manage projects or how people cooperate may account for them.

Ideally, the partition among team members would be equal. However, Figure 8 displays an unexpected result. In group1 and group2, many issues were not assigned to specific person, yet in group3 most issues were assigned to user9. A possible explanation for group1 and group2 is that those issues without assignee were tasks requiring all members to work together. For group3, the result may be a waning in teamwork project since most issues were assigned to a specific person. However, there still exist possibilities that issues assigned to user9 are more granular than others or only user9 managed issues with Github.

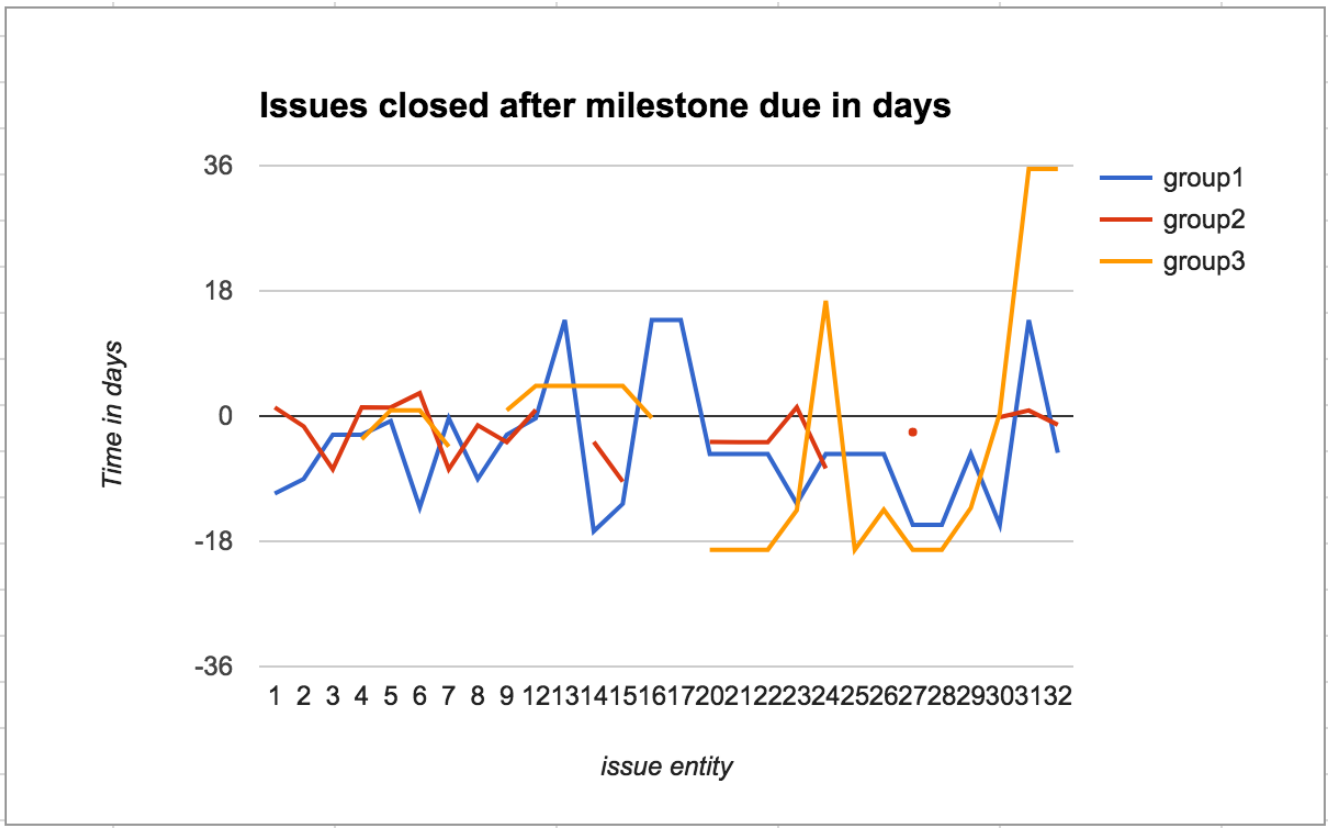

Figure 6. Issue closed after milestone due

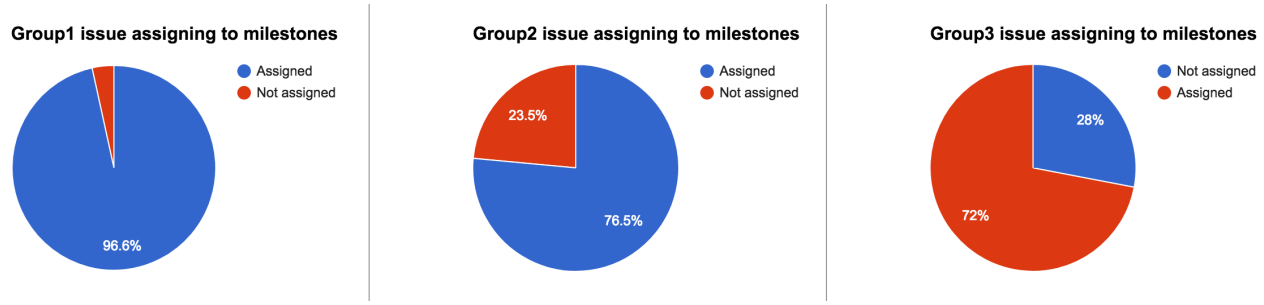

Figure 7. Issue assigned to milestones

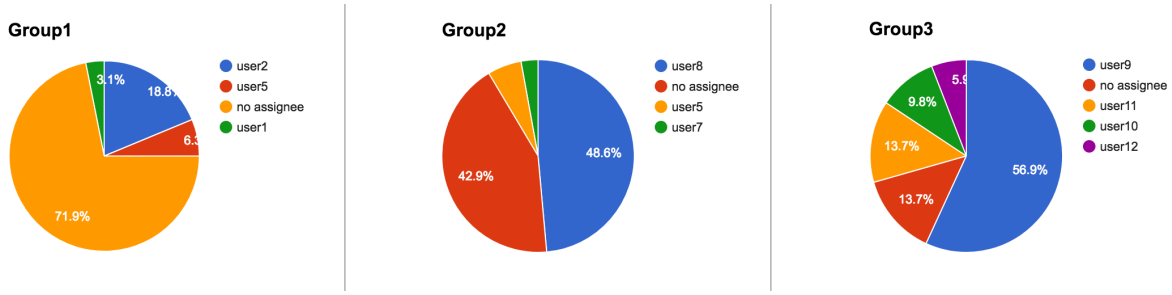

Figure 8. Issue assigned to specific member

4 Commits

Commits where another piece of data that we gathered to analyze for groups H, P, and S. The data for commits is broken down by person, timestamp and the title on of the commit. For the purposes of this paper we decided to ignore the title of the commits and focus on the relationships between the person making the commit and the timestamp of the commit. Specifically we analyzed the frequency of commits by individual team members and how they changed during the course of the project. Our initial thought was that we would be able to identify group dynamics and individual work patterns by analyzing when group members made commits.

Figure 9. Commit statistic for group H

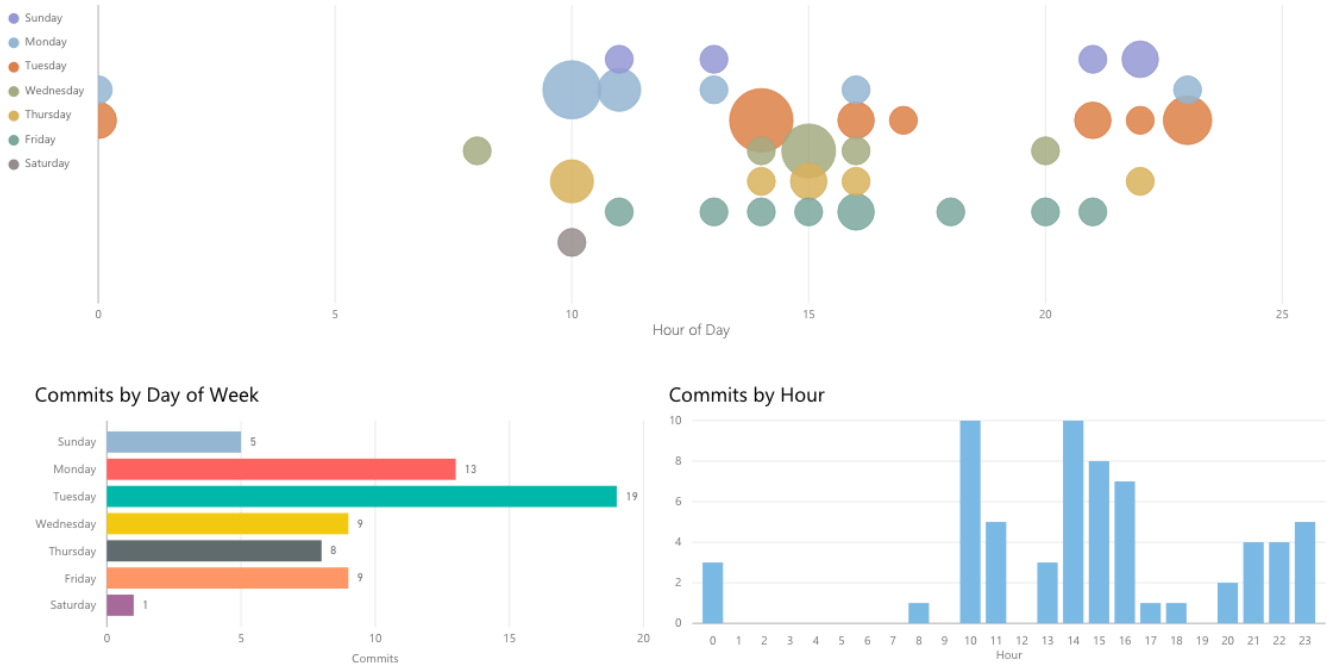

Figure 10. Commit statistic for group P

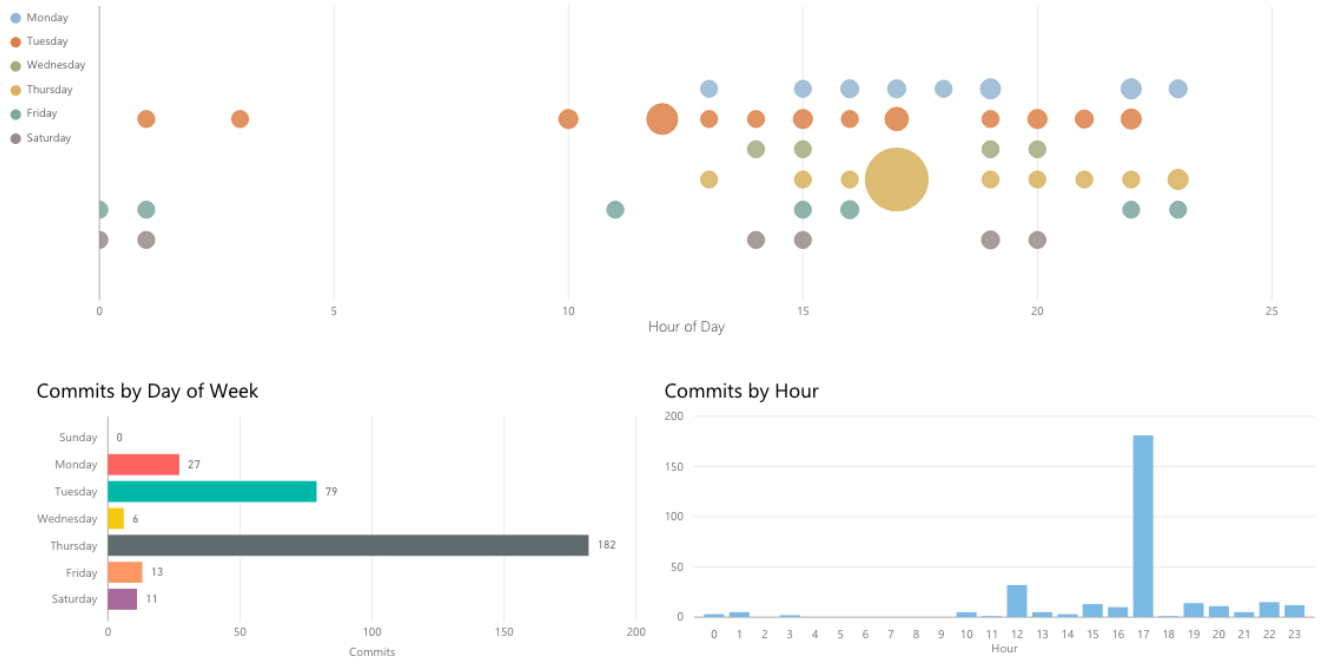

Figure 11. Commit statistic for group S

4.1 Hypothesis-Questions

Our first question was, if a group member made lots of commits within the first month of the project, did they continue that pace throughout the project or did they slowly reduce their number of commits as the project progress? Our initial hypothesis was that group members who made the majority of the commits earlier in the project may be less likely to make commits at the end of the project. This was thought to be a typical student behavior. To reduce their work load, students put lots of effort into the initial phases of a project knowing they can reduce their efforts towards the end of the project as the other group members feel the need to contribute more.

Our second question was, could we determine a group leader by analyzing the number of commits through the three phases of the project? Our hypothesis was that a group leader would submit more commits in the first phase of the project as they organized the group meetings and gave guidance to the team. During the second phase of the project we would expect to see a fairly even distribution of commits by each member. These commits would include the code that each member committed as well as any internal dialog as to how their part was progressing. Finally in during the last phase of the project we would expect that the group leader is making more commits than the other group members as they finish up the final details before submitting the project.

4.2 Commit Data Anomalies

During our data processing we discovered two data anomalies that will affect our findings. First we found that group S had a disproportionately higher number of commits than group H and P. However as we looked deeper into this anomaly we discovered that 85 commits were made by one group member on the same day and within minutes of each other. After diving deeper, we discovered that this group member was deleting files within GitHub and it was registering as a valid commit in the data that we pull. To normalize group S's commits with group P and H, we decided to disregard the 85 commits from our data. The second data anomaly we discovered was that group H made 95% of their commits in January which left zero commits for February and only three commits combined in March and April. The labels for their commits during January were constant with what would be anticipated for the January project. We suspect that either we made a mistake in pulling their data from GitHub or they completed most of their work on a different or personal GitHub repository that we did not pull from.

4.3 Commit Analysis

From the three groups analyzed, there is evidence that one person does make the majority of the commits during the first phase of the project. (Figures [] shows January Commits) In group P and S person one submitted 100% of all commits for January. However, this did not indicate a decrease in overall commit level during the phase 2 and 3 of the project. In group S, the group member who committed 100% of phase one achieved a commit level of 50% during phase two and 24% during phase 3.

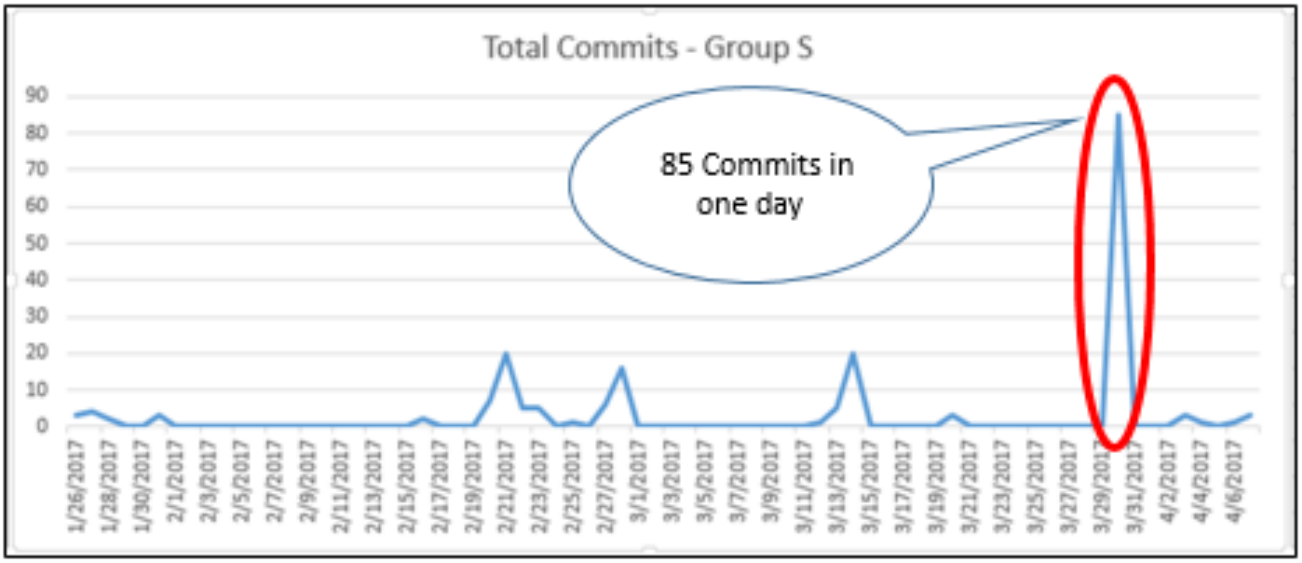

Figure 12. Group S Commit Anomaly

Although this group member dropped to 24% of the commits in phase 3 it can be explained by group member 3 submitting 17 delete commits in one day; this is the same group member who submitted 85 delete commits on the same day which were removed from the dataset. In group P, the member who submitted 100% of commits in phase one achieved a commit level of 58% during phase 2 and 71% during phase 3. It is also worth noting that in both group S and group P the users with 100% of phase one commits where the users that submitted the last commits in phase three which were labeled "adding final paper" for group S and "User analysis paper" for group P. Group P and S's commit level is indicative of having a group leader who gets the group organized at the beginning of the project, gives guidance during the project and finally makes sure the final paper is submitted. Although it's difficult to know for sure, our second hypothesis seems like a reasonable assumption that the person who makes the most commits during phase one is likely to be the team leader. This assumption may be useful to assessing the success or failure of a project due to poor leadership and guidance.

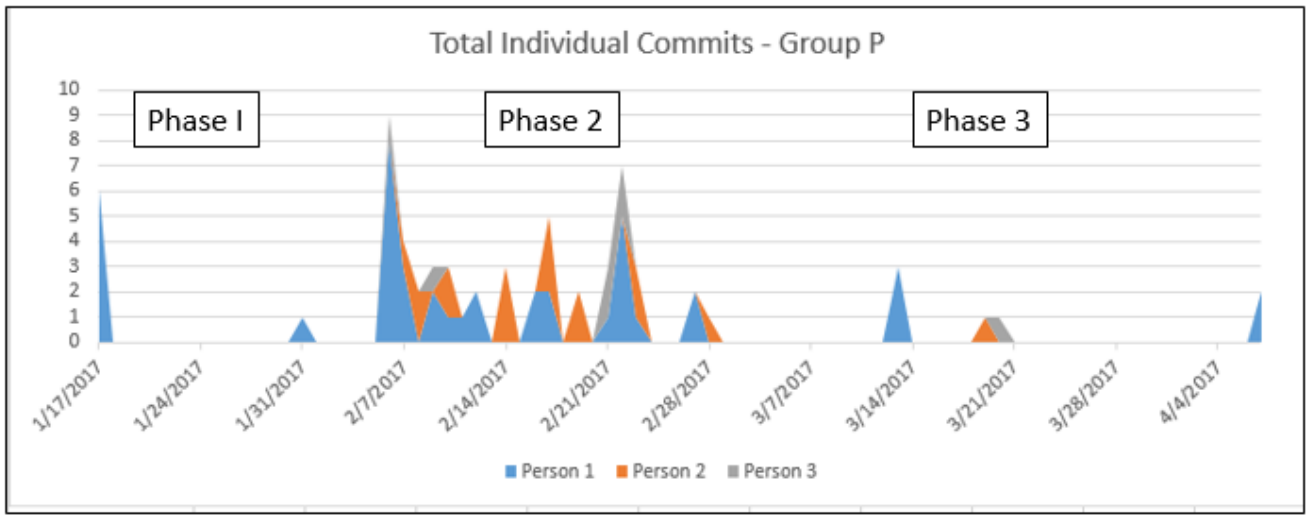

Figure 13. Total Individual Commits - Group P

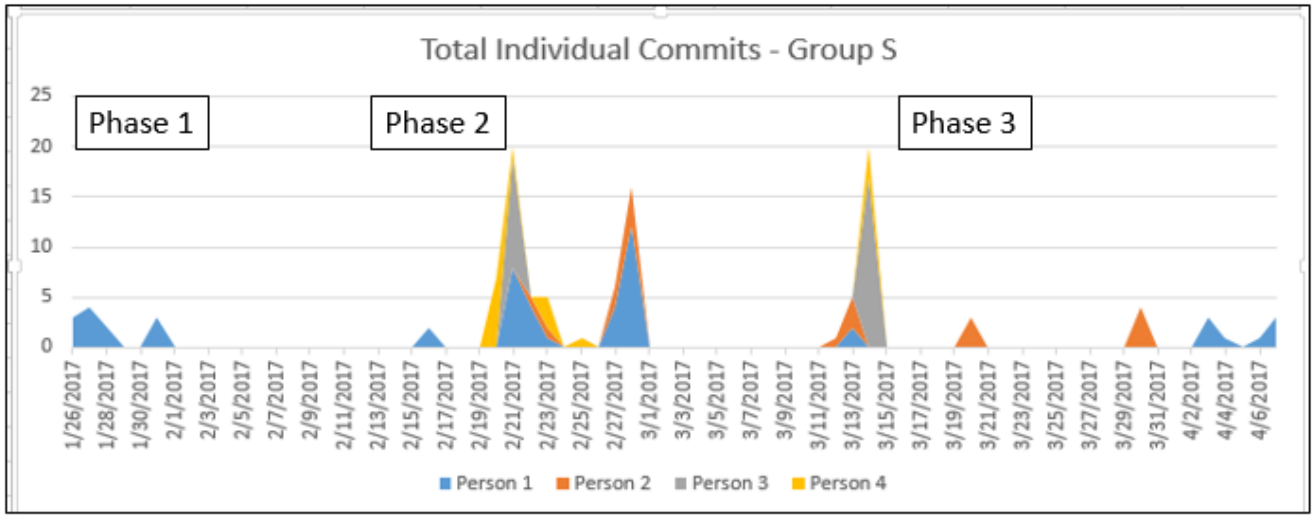

Figure 14. Total Individual Commits - Group S

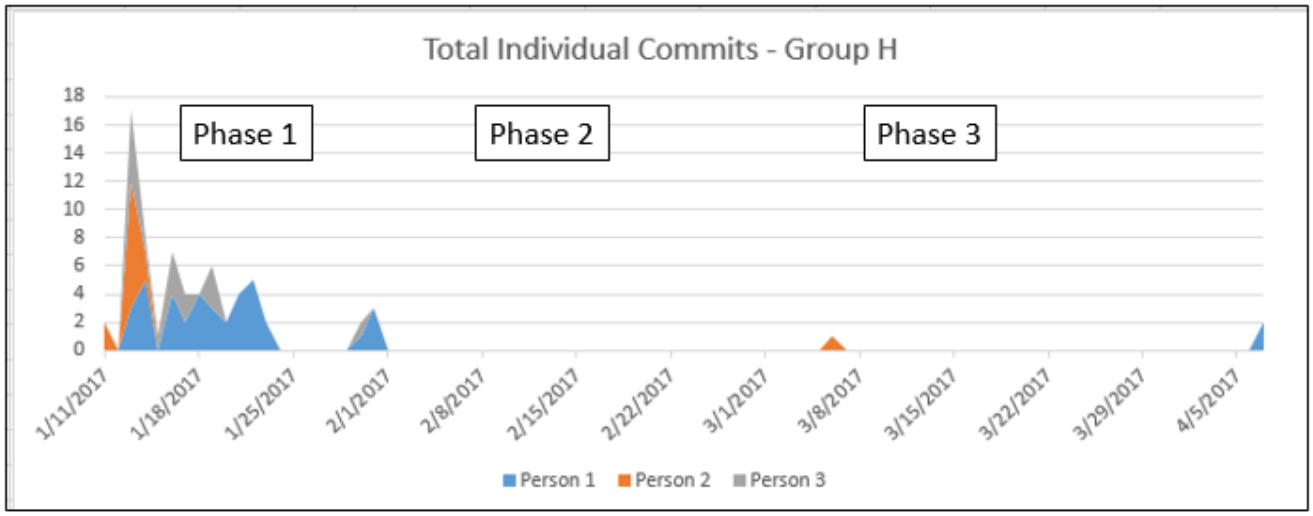

Figure 15. Total Individual Commits - Group H

5 Project comments

Most users follow a common pattern - the size of the circles - in discussing matters except for 3 users. User 7 is very vocational while user 2 and 5 were quite introvert.

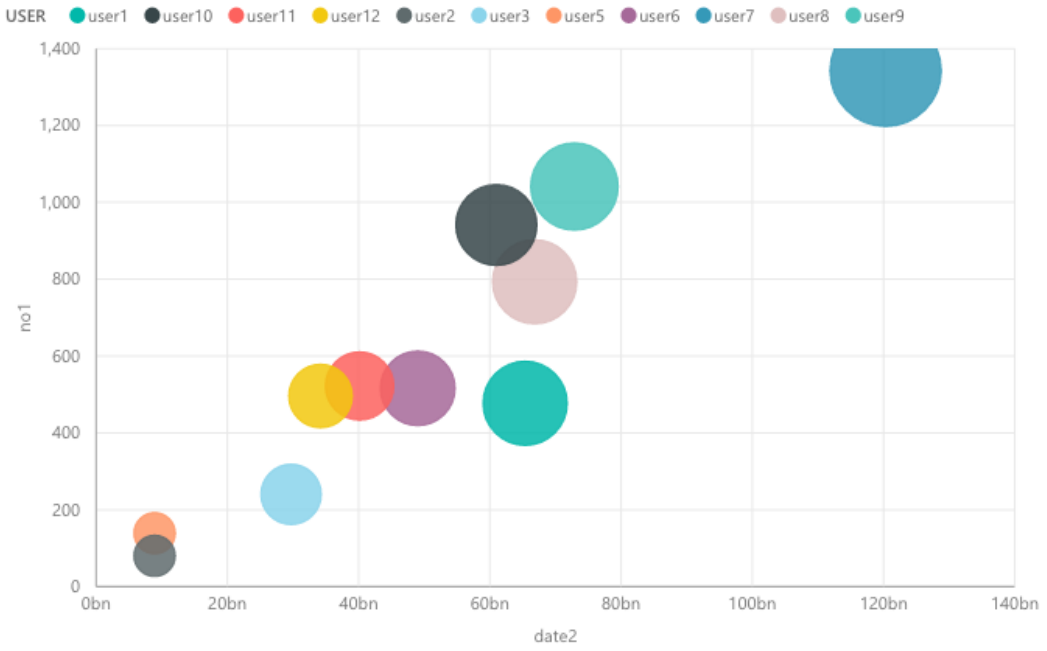

Figure 16. Comment distribution - users from all 3 groups

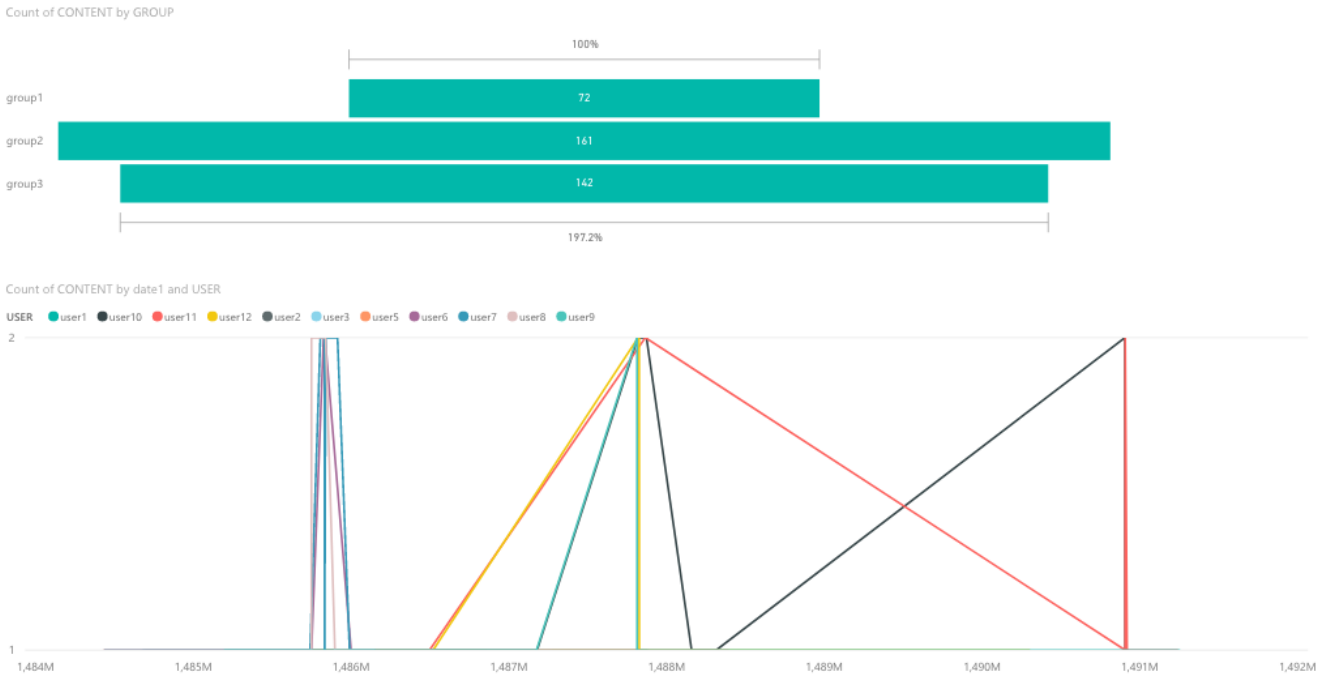

Figure 17. Comment distribution by group and by progression of time

Group 1 has much less interactions between members compared to other groups. The reason is probably they only post comments when they were reminded by the Professor (and so did other groups as shown on time distribution). Members in group 2 and 3 communicate with each other on another important milestone (the second converge in time distribution graph). User 2 and 11 communicated until the end of the project. All groups have the same very low standard deviation. Group 3, however, has a significant high variance. Comparing the results with other figures, it appears that group 3 has the most healthy pattern of member communication. Their communication traffics have a decent volume but are spread out.

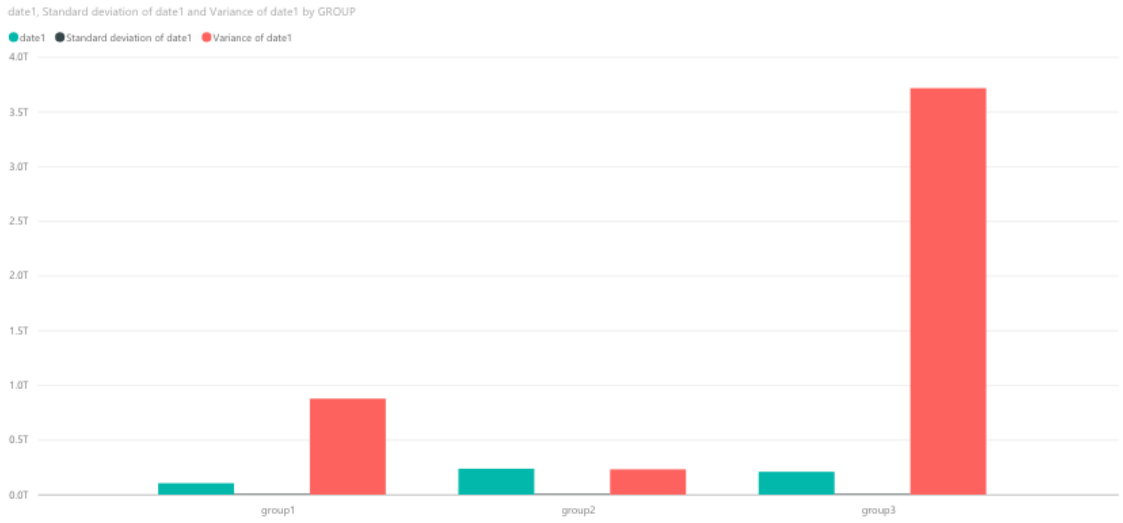

Figure 18. Comment statistics based on time

Figure 19. Communication distribution among members per group

There are differences in the way members of each group participate in discussions. The second group has a very active member who posted more than half of the group's comments. That can be a good thing if the member was active in replying to other members' questions. Participation rate is more balanced within the third group. Their discussions are also very consistent, spreading out over a good period of time. For some reasons, first group had an outstanding conversation toward the end of the project where most other groups didn't have to.